Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Вятский государственный университет»

Колледж ВятГУ

**ОТЧЕТ**

**ПО ДОМАШНЕЙ КОНТРОЛЬНОЙ РАБОТЕ №5**

**«ИССЛЕДОВАНИЕ АЛГОРИТМОВ СОРТИРОВКИ»**

**ПО ДИСЦИПЛИНЕ «ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ»**

Выполнил: студент учебной группы

ИСПк- 101-51-00

**Дуркин Арсений Валерьевич**

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2022

В отчете должны отображаться:

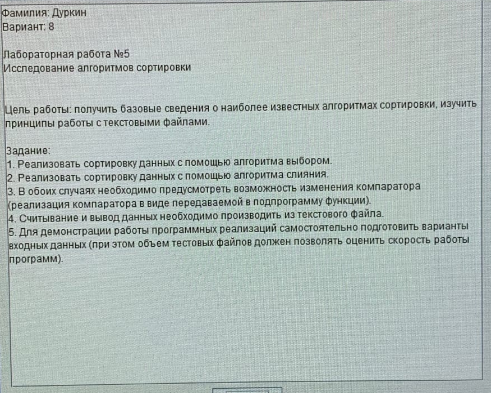
1. Цель лабораторной работы
2. Формулировка задания (с вариантом)
3. Схема алгоритма с комментариями
4. Код программы
5. Результат выполнения программы
6. Вывод
7. Цель: получить базовые сведения о наиболее известных алгоритмах сортировки, изучить принципы работы с текстовыми файлами.
8. 

Рисунок 1 - Задания

**uses** crt, Utils;

**var**

f: text;

**var**

a: **array** [0..99999] **of** integer;

kolelem, viborsort, komparator, p: integer;

**begin**

Assign(f, 'ДКР 5.txt');

reset(f);

kolelem := 0;

Read(f, kolelem);

write(kolelem);

**for var** i := 0 **to** kolelem - 1 **do**

**begin**

Read(f, a[i]);

write(a[i], ' ');

**end**;

writeln;

writeln;

writeln('Способ сортировки. Чтобы отсортировать выбором, нажмите 1. ');

writeln('Чтобы отсортировать вставками, нажмите 2.');

read(viborsort);

writeln('Выбор компаратора. Чтобы отсортировать по возрастанию, нажмите 1. ');

writeln('Чтобы отсортировать по убыванию, нажмите 2.');

read(komparator);

**if** (komparator = 1) **or** (komparator = 2) **then**

**begin**

MillisecondsDelta;

**case** viborsort **of**

1:

**begin**

**for var** i := 0 **to** kolelem - 2 **do**

**for var** j := i + 1 **to** kolelem - 1 **do**

**if** ((a[j] < a[i]) **and** (komparator = 1)) **or** ((a[j] > a[i]) **and** (komparator = 2)) **then**

**begin**

p := a[i];

a[i] := a[j];

a[j] := p;

**end**;

**end**;

2:

**begin**

**var** i, key: integer;

**for var** j := 1 **to** kolelem - 1 **do**

**begin**

key := a[j];

i := j - 1;

**while** (i >= 0) **and** (((a[i] > key) **and** (komparator = 1)) **or** ((a[i] < key) **and** (komparator = 2))) **do**

**begin**

a[i + 1] := a[i];

i -= 1;

**end**;

a[i + 1] := key;

**end**;

**end**

**else**

writeln('Введено некоректное значение сортировки. Завершение программы');

**end**;

**if** (viborsort = 1) **or** (viborsort = 2) **then**

**for var** i := 0 **to** kolelem - 1 **do**

write(a[i], ' ');

writeln;

writeln;

writeln('Время выполнения программы ', MillisecondsDelta);

writeln;

writeln;

**end**

**else**

writeln('Введено некоректное значение компаратора. Завершение программы');

**end**.

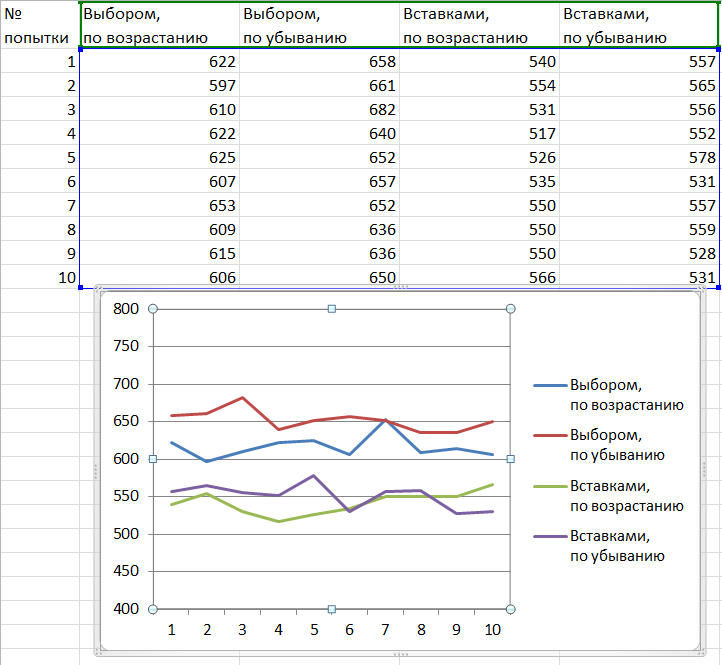


Рисунок 2 – График выполнения времени программы

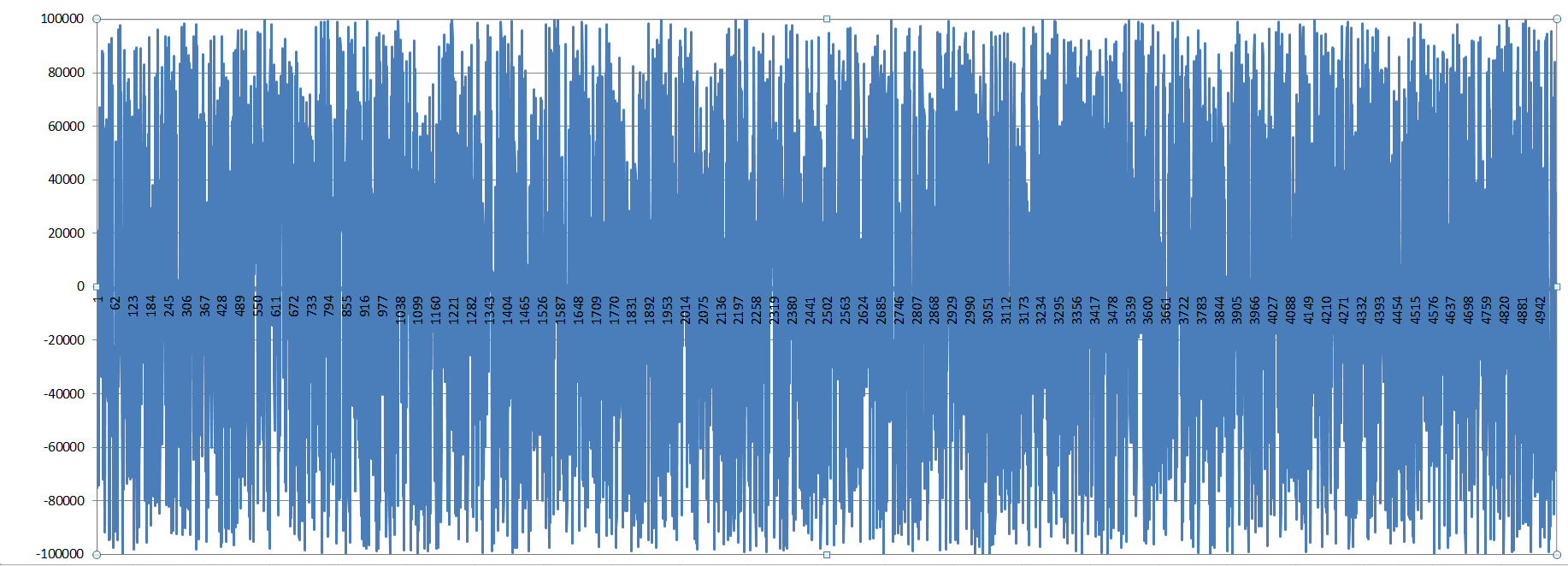


Рисунок 3 – График

6. В ходе работы была получена программа, сортирует числа из файла по возрастанию и по убыванию.

В ходе работы были получены знания о Здоровье Паскаль, как оптимизировать код на Паскале так, чтобы его уровень здоровья был 100% и выше. Программы написаны без использования встроенных функций. Была освоена сортировка разными методами. Была освоена программа, которая подсчитывает количество времени выполнения программы. Было освоено считывание и вывод данных из текстового файла.

В результате выполнения программы, получена среднее количество времени выполнения 1 и 2 сортировки по убыванию и возрастанию, в виде графиков.